

Weekly Report(March.24,2019-March.31,2019)

Rui Shaopu

Abstract

This week I only learned RNN and LSTM since there are much homework this week.

1 Learned

After finishing all the thing about them, I have to say, it's easy to finish code if you know the formula. But it's hard to understand how they work and why they can make it.

1.1 RNN

1.1.1 Introduction

Recurrent Neural Network. It's a typical model to address sequential data, such as videos, sentences, music and so on.

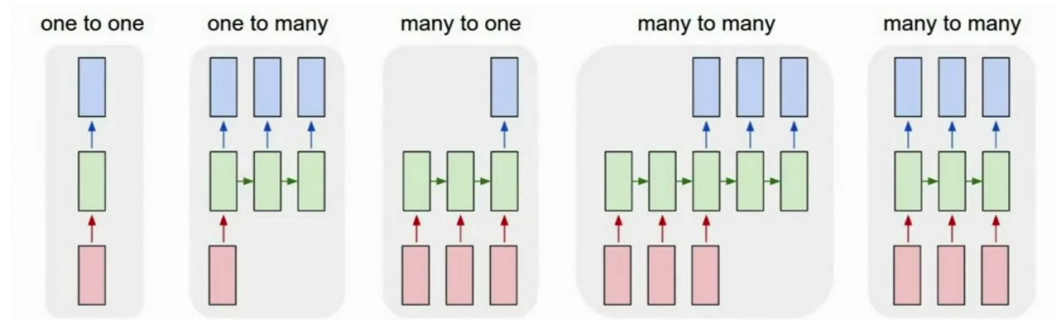


Figure 1: Different kinds of RNN

Formula:

for each hidden layer:

$$h_t = f_w(h_{t-1}, x_t) = \tanh(W h_{t-1} * h_{t-1} + W x * X + b)$$
$$y_t = W_{hy} h_t$$

h_t is the new state, while h_{t-1} is the old state, and x_t is input of this timestep. f_w is some function with parameters W , the right part of the equation is the function used in homework. y_t will be used when calculating loss.

1.1.2 My work

Actually It's much more easier than other parts, so I will just show the result.

RNN: Computational Graph: Many to Many

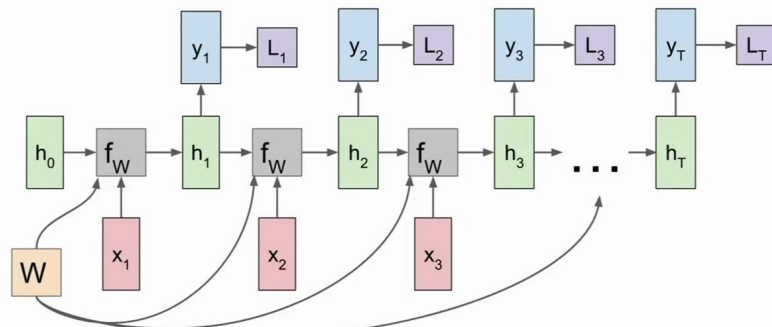


Figure 2: Typical many-to-many model

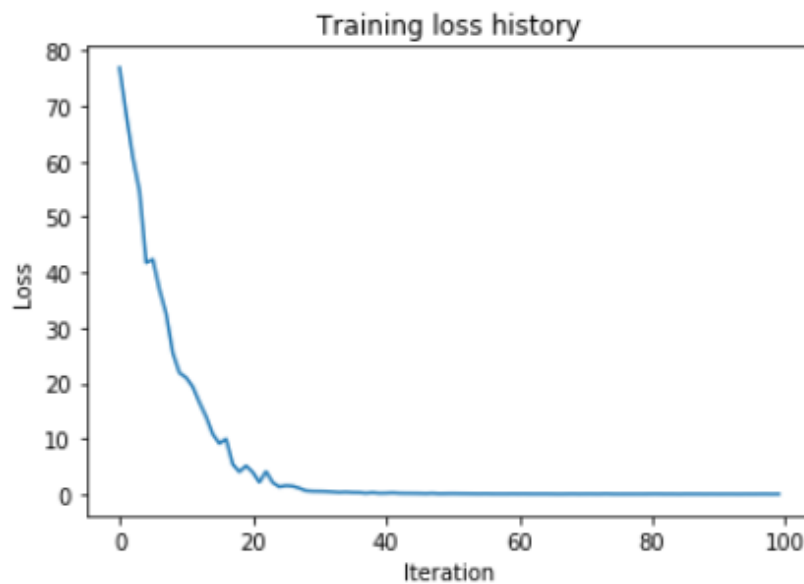


Figure 3: loss

train
 <START> a boy sitting with <UNK> on with a donut in his hand <END>
 GT:<START> a boy sitting with <UNK> on with a donut in his hand <END>



train
 <START> a man <UNK> with a bright colorful kite <END>
 GT:<START> a man <UNK> with a bright colorful kite <END>



Figure 4: trained result

1.2 LSTM

1.2.1 Introduction

Long Short Term Memory, is slightly fancier recurrence relation for RNN. It's designed to solve the problem of vanishing and exploding gradients.

Formula:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

In this formulas, i means *input_gate*, f means *forget_gate*, o means *output_gate*, while g doesn't have a good name.

1.2.2 My work

The implementation is the same as RNN, so I will still say nothing about it.

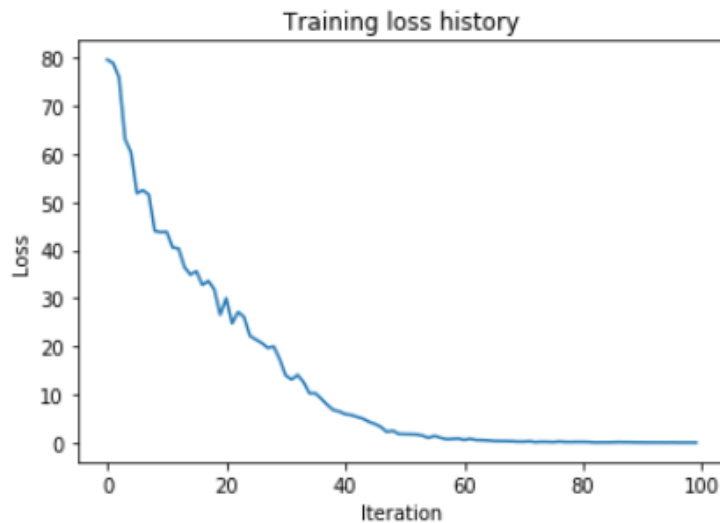


Figure 5: Final result

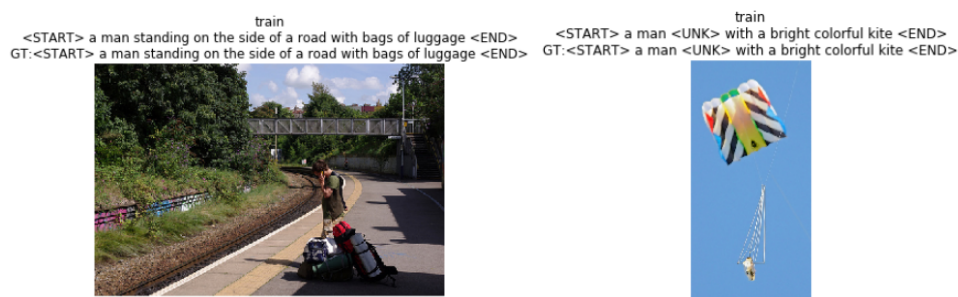


Figure 6: trained results

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

2 Understanding

RNN addresses sequential words in the homework. In order to predict next layer's state, it will use all of the previous state as input. It works very well. However, it has the problem of vanishing and exploding gradients. That's why LSTM exists. LSTM will forget part of previous hidden state, which makes the network work better.

Question: As training goes on, I come to confuse about the relation between an actual layer and its functions. For example, why adding convolutional layers in a CNN will get better performance? I know there must be some scientific basis, so how to use this basis if I want to improve a model?

3 Plan

Keep going.